3. (Original) The method according to claim 1 wherein the instruction operand is a pointer to the number.

4. (Currently amended) The method according to claim 3[1], wherein the pointer is stored in a register.

5. (Original) The method according to claim 1, wherein the operand identifies a register that stores the number.

6. (Original) The method according to claim 1, further comprising loading the number into a register.

7. (Original) The method according to claim 6, further comprising changing the number stored in the register based on processor cycles until the number reaches a predetermined value.

8. (Original) The method according to claim 7, wherein the changing is decrementing the number stored in the register.

9. (Original) The method according to claim 7, wherein the changing is incrementing the number stored in the register.

10. (Original) The method according to claim 7, wherein the predetermined value is zero.

11. (Original) The method according to claim 7, further comprising generating an interrupt disable signal during the changing of the number.

12. (Original) The method according to claim 11, further comprising generating an interrupt enable signal when the number reaches a predetermined value.

13. (Original) The method according to claim 7, further comprising writing the register with a value based on a write instruction.

14. (Original) A processor including an interrupt disable instruction processing feature, comprising:

a program memory for storing instructions including an interrupt disable instruction having an operand specifying a number corresponding to an interrupt disable duration;

a register for storing the number;

an instruction fetch/decode unit for fetching and decoding instructions, the instruction fetch/decode unit decoding the interrupt disable instruction and disabling the interrupt processing capability of the processor based on the number.

15. (Original) The processor according to claim 14, wherein the register changes the number based on processor cycles.

16. (Original) The processor according to claim 15, wherein the register changes the number by incrementing it.

17. (Original) The processor according to claim 15, wherein the register changes the number by decrementing it.

18. (Original) The processor according to claim 15, wherein the [predetermined value is] disabling the interrupt processing capability ends when the number reaches zero.

19. (Original) The processor according to claim 15, further comprising:

interrupt disable logic, coupled to the register, the interrupt disable logic generating an interrupt disable signal during the changing of the number.

20. (Currently amended) The processor according to claim 15, wherein the interrupt disable logic generates an interrupt disable [enable] until the number reaches a predetermined value.

21. (Original) The processor according to claim 15, wherein a write instruction updates the register.

REMARKS

The applicants have carefully reviewed the office action. In response, the applicants have amended claims 1, 4, 18 and 20. In the office action, the Examiner rejected claim 20 under 35 U.S.C. §112, first paragraph. The amendment to claim 1 highlights that the interrupt disable instruction is fetched from an instruction sequence. The applicants have amended claim 20 to focus on generating an interrupt disable as opposed to an interrupt enable and believe that this eliminates the basis for the Examiner's rejection. The examiner also rejected claims 4 and 18